DECLARATION UNDER 37 C.F.R. § 1.132
OF MAURICE CASTRO, PH.D.

1. I am a Computer Scientist holding a degree of Bachelor of Science (Hons.) from Monash University, Melbourne, Australia, and a Ph.D. in Computer Science, also from Monash University. I have experience in a number of areas of computer science, including software engineering, operating systems and computer architecture. A copy of my *curriculum vitae* is attached hereto and marked Exhibit MC-1.

2. Between March 1999 and March 2003 I worked at the Software Engineering Research Centre (SERC), and during that time was familiar with the work conducted by my colleagues Professor Fergus O'Brien and Dr. Matthew Roughan ("the inventors") in relation to scalable computer systems.

3. The name given to the massively scalable computer system that is now the subject of US Patent Application Serial No. 10/030,342 ("the application") was "Magnus" for the smaller "building blocks" or clusters, or "Magnum" when referring to a larger system having clusters interconnected in accordance with the principles of the invention claimed in the application. I personally worked on an emulated system, to demonstrate the feasibility of writing software for the "Magnus Machine" in the Erlang programming language. My work confirmed the possibility of achieving the performance improvements predicted by the provision of a massively scalable architecture in accordance with principles of the invention. In particular, this work confirmed the prediction that near-linear speedup in application performance can be achieved in proportion to the number of available processors. A copy of a research report that I co-authored in relation to this work, entitled *"Magnus: Software Demonstration"* and published as SERC Technical Report 119 in April 2000 by the Ericsson Software Engineering Research Centre, RMIT University, Melbourne Australia is annexed hereto and marked Exhibit MC-2.

4. I have reviewed the presently pending claims of the application, a copy of which are annexed hereto and marked Exhibit MC-3.

5. I have also been provided with, and have reviewed, copies of three documents that I have been informed have been cited against the application. These three documents are Marc

Barthelemy *et al "Small-World Networks: Evidence for a Crossover Picture"* (April 12, 1999), James J Collins *et al "It's a Small World"* (June 4, 1998), and Clement R Attanasio *et al,* US Patent No. 5,371,852 (December 6, 1994). Hereafter, these documents are identified as "Barthelemy," "Collins," and "Attanasio."

6. I initially note that Barthelemy, Collins, and Attanasio clearly do not, alone or in combination, disclose a method for constructing a scalable computer system or a scalable computer system as defined by either of claims 8 or 19. However, I understand that it is being asserted that it would have been obvious to apply mathematical principles disclosed in Barthelemy and Collins in order to arrive at the claimed invention based on specific practical applications suggested in each reference. It is my opinion that it would not have been obvious, in June 1999, for a person of ordinary skill in the art to have arrived at the inventions defined by either of claims 8 or 19 of the application based upon the combined teachings of Barthelemy, Collins, and Attanasio. My reasons for holding this opinion are set out in the following paragraphs.

7. Barthelemy and Collins essentially relate to a mathematical concept of a small-world network. In mathematics, the term "network" refers to a graph having nodes that are interconnected by edges. Each edge may have an associated direction and/or weight. Importantly, while mathematical networks may be used to model real-world networks, it is not conversely true that all mathematical networks can be converted into a corresponding implementation of some form of real-world network. A mathematical network is an abstract construct, whereas real-world networks of various types (including computer interconnects and telecommunications networks) are subject to design constraints, such as cost, performance, power consumption, and other requirements.

8. It is my belief that Barthelemy and Collins are examples of publications dating from around 1998/1999 that introduced a mathematical concept of a small-world. Upon information and belief, these references predate any corresponding computer or communications networking concept of a small-world.

9. Barthelemy and Collins include comments regarding the potential applicability of the mathematical concept of the small world to real-world networks. In particular, Barthelemy

suggests that its results may be applicable to future studies of flow in information networks (see page 3183, column 1), while Collins suggests that small-world principles may be applied to improving the performance of cellular phone networks or the Internet (see page 410, column 1). However, for at least the reasons set out in the following paragraphs, I do not believe that these statements would or could, as of June 1999, lead a person having ordinary skill in the relevant art to the inventions defined by the claims of the application. Moreover, as of June 1999, I do not believe that the practical applications suggested by Barthelemy and Collins could have been heeded by one of ordinary skill in the art with any reasonable expectation of success.

10. Firstly, it is my opinion that the relevant art in the present case is that of computer architecture generally, and supercomputers in particular. I note in this regard that claim 8 is directed to "a method for constructing a scalable computer system," while claim 19 is directed to "a scalable computer system." Barthelemey Collins and Attanasio address only telecommunications and computer networks – systems which are loosely coupled. Persons knowledgeable in the art would consider the system described and claimed as tightly coupled and hence in the domain of computer architecture.

11. Tightly-coupled multiprocessor and supercomputing systems are characterized by the use of dedicated high-performance internal interconnects for communication between processors, and may be distinguished from "loosely coupled" systems comprising a large number of computers and/or processors interconnected using conventional data-networking technologies, such as Ethernet. The tight time constraints on tightly-coupled multiprocessor systems prevent the use of many solutions often used in the fields of telecommunications and data networking. There is a strong division between the topics of computer architecture (interconnects) and computer networking (communications networks). It is therefore my opinion that there is no suggestion in either Barthelemy or Collins that the principles of (mathematical) small-world networks may be applied to the design and construction of improved tightly-coupled scalable multiprocessor computing systems so as to render obvious the inventions defined by claims 8 and 19.

12. Secondly, it is my opinion that, at least as recently as June 1999, the practical applications

suggested by Barthelemy and Collins could not have been attempted by one of ordinary skill in the art with any reasonable expectation of success. Barthelemy and Collins were published, respectively, in the journals *Physical Review Letters* and *Nature*. These journals are not known for the publication of articles relating to computer systems architecture or engineering. Based upon my reading of Barthelemy and Collins, I do not believe that their authors had relevant specialist knowledge in the field of computing or telecommunications. In my opinion, the suggestions in the references that their mathematical results may be applied to improvements in telecommunications and data-communications networks are speculative, almost in the nature of science fiction, in that they posit possible ultimate benefits and outcomes of the mathematical results without providing any teaching as to how such outcomes might be achieved in practice. It is my opinion that, at least as recently as June 1999, there were significant technical and engineering barriers to overcome in order to apply the small-world principles to real-world networks, and that there were no obvious solutions to these problems.

13. Even bringing to bear the relevant knowledge of a person having ordinary skill in the art of data-communications or telecommunications, it is my further opinion that such a person would not, in any event, treat the suggestions in Barthelemy and Collins seriously, due to the fact that these documents completely ignore the practical realities applicable to such networks at the relevant time (*i.e.*, June 1999).

14. Although real-world communications networks (including the Internet and cellular and telephony networks) are technically graphs, the majority of their structure is implemented as physical and logical trees. Such networks are characterized by non-uniformity, *i.e.*, they comprise hierarchies having high-capacity roots and trunks that interconnect ever-increasing numbers of lower-capacity branches. Internet-routing protocols are typically distance-vector algorithms. These protocols excel when dealing with trees; however, when dealing with graphs they are subject to pathological cases as they do not take link capacity into account. The introduction of weights, typically by hand, is required to correct pathological cases. Careful planning of real-world networks is required to ensure that there is adequate capacity. Where in-band signaling is used (*i.e.*, Internet Protocols), a failure in the capacity of a relatively few links near the core of the network can result in widespread

disruption due to route flapping.

15. The approach suggested by Collins of introducing "a few random connections" between cells in a cellular network, or between "nodes along the backbone of the Internet" is therefore simply not workable, and would be recognized and dismissed as such by a person having ordinary skill in the art. Additional links of this type could not be effectively utilized without making corresponding unspecified modifications to routing protocols and would not address the problem of providing overall improvements in performance throughout the network (*i.e.*, in the branches of the tree as well as in the trunk). Worst yet, the introduction of additional "random links" would create multiple paths between nodes of the network, which to make effective use of their capacity would require an ambiguity of routing of data packets and hence could lead to routing cycles, whereby data packets may circulate around pairs of alternative routes and the need for larger packet reconstruction buffers to resequence out of order packets. These types of problems are generally avoided in real networks through the use of "spanning tree" algorithms in the routing nodes, which effectively ignore alternative routes and impose a "logical tree" topology over the underlying network.

16. Even if the approach suggested by Collins were to be accepted at face value, it would not result in the inventions defined by claims 8 or 19. More specifically, introduction of "random links" in the pre-existing communications networks would not result in those networks exhibiting small-world properties as set forth in claims 8 and 19. At best, the new links would provide redundancy and/or additional capacity that might slightly improve the reliability and/or efficiency of the network in accordance with well-known principles, but without the emergence of any small-world characteristics.

17. I further note that Barthelemy and Collins utilize one-dimensional (*i.e.*, linear or ring) networks as their starting points, and posit that their results may be applied to real-world communications networks. However, as already noted, such networks are not one-dimensional, linear, or ring networks. Furthermore, it is not apparent how other types of known communications networks that are based on linear or ring topologies could be adapted to utilize small-world principles, since the access protocols employed in such

networks are closely coupled to their physical topology and would not function if the topology were to be changed. Indeed, embodiments of the invention described in the application do not start out as a one-dimensional, linear or ring network. Thus, one of ordinary skill in the art in June 1999 would not have a reasonable expectation of success for the method of rewiring proposed by Barthelemy and Collins. Moreover, the method of rewiring is not applicable to the inventions claimed in claims 8 or 19.

18. Additionally, Barthelemy and Collins are completely silent on the topic of routing protocols, since the primary concern of each document is with the mathematical properties of small-world networks.

19. Furthermore, Barthelemy and Collins are silent on the subject of scalability. Each document discusses the mathematical properties of networks having a given number of nodes. Scalability is a significant issue in communications networks and data interconnects, and the scaling of networks such as the Internet and cellular telephony networks is generally difficult and costly. In particular, as such networks grow, typically by the addition of further nodes in the "branches" of the tree, it becomes necessary to upgrade the processing and transmission capacity in the larger branches, trunk and root of the tree in order to support the additional traffic.

20. I would note that, upon information and belief, in the decade since the publication of Barthelemy and Collins, no one has implemented their suggested modifications to existing communications networks in order to provide improved performance. I believe that this is because, as I have explained in detail above, such modifications are simply impractical, result in systems deficient from those claimed in claims 8 and 19 and, moreover, would not work. In my opinion, one of ordinary skill in the art in June 1999 would not have had any reasonable expectation of success for the suggested modifications. Furthermore, as far as I am aware, the inventors are the only ones to have recognized the potential of the small-world principle for the design and construction of massively scalable multiprocessing computer systems, or supercomputers.

21. Attanasio discloses a method and apparatus for enabling a cluster of locally interconnected computers to appear as a single node on a network. I understand that the main objective of

Attanasio's invention is to enable a cluster, and/or the computers comprised by a cluster, to be reconfigured without having to modify or notify any computer outside the cluster. This approach is applicable to loosely coupled distributed computing systems. In any event, I am unable to identify any disclosure within Attanasio that is relevant to the design and/or construction of scalable computer systems of the type defined by the claims of the application.

22. In contrast to Barthelemy, Collins, and Attanasio, when considered as a whole, the presently claimed invention combines prior art elements in a completely new way, achieving an unexpected synergy that results in a scalable multiprocessing computing system that is able to support from a few nodes up to hundreds of millions of processing nodes with no appreciable communications or performance limitation. More particularly, claims 8 and 19 reference a scalable computer system and recite an architecture in which each added node (*i.e.*, processing unit) along with its associated link to the network provides all of the additional processing and transmission capacity required in order to support the additional node. For example, claim 8 requires that steps of providing a plurality of cross-links and directly connecting a plurality of pairs of node clusters in accordance with a selection process be repeated until the resulting network comprises a small-world network having an average path length between a plurality of computing nodes falling within a predetermined desired range, independently of a number of the plurality of computing nodes. In similar fashion, claim 19 requires that a network comprise a small-world network having an average path length between nodes falling within a predetermined desired range, independently of a number of a plurality of computing nodes.

23. The suggestions of Collins and Barthelemy do not address providing a scalable, small-world network as required by claims 8 and 19. Indeed, none of the prior art referenced addresses the scalability of a network. At most, they address methods of making a network perform better at a given size rather than independently of a number of a plurality of computing nodes as recited by claims 8 and 19. At least by June 1999, the inventors, for the first time, conceived that small worlds could be used as described in claims 8 and 19 to construct a small-word network having an average path length between a plurality of

computing nodes falling within a predetermined desired range independently of a number of the plurality of computing nodes. Moreover, at least by June 1999, the inventors conceived, for the first time, that remarkable scalability could be achieved thereby. For example, it was found that small worlds could be used as described in claims 8 and 19 to construct a computer system, using the same components, that spans over five orders of magnitude. This result was surprising. Super computers in June 1999 and to a large extent now are designed with a particular size in mind when they are constructed. The components are selected to support that size. Being able to span one order of magnitude shows considerable forethought and is typically achieved by populating pre-prepared places in the initial design. The scalable computer system as set forth in claims 8 and 19 allows for organic growth of the computer and its performance independently of a number of computing nodes. This result was surprising and unpredictable to me and, in my opinion, would have been surprising and unpredictable to anyone of ordinary skill in the art in June 1999.

24. I am aware that this declaration is being submitted to the United States Patent and Trademark Office with reference to the application. I declare that all statements made by me in this declaration of my own knowledge are true and that all statements made on information and belief are believed to be true. I declare further that the statements made by me in this declaration are being made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that any such willful false statements made by me in this declaration may jeopardize the validity of the application or any patent issuing thereon.

14/9/2009
Date

Maurice Castro
Dr. Maurice Castro

EXHIBIT MC-1

MAURICE CASTRO
3/26 Powell St,
South Yarra 3141 Australia
+61 (0)3 9867 1390
+61 (0)414 867 139
maurice@castro.aus.net

PROFILE

Deep technical knowledge of computer science, software engineering, networks and telecommunications, has been the core of my career. For over 8 years my focus has been entrepreneurship and research and development to produce exciting projects and businesses. These projects were in the areas of computer security, Internet infrastructure and computer architecture. Experience in academia provides the ability to explain, present and teach complex technical concepts. More recently I have been actively involved in the management of an SME delivering quality Internet services and a wireless telecommunications carrier.

HIGHLIGHTS

- Developer

    Developed a wide range of projects and environments. The projects include a compiler for the Erlang programming language and one of the original load balancing / admission control system for web servers. Environments range from kernel and device driver programming through to parallel systems and from system programming through to application programming.

- Entrepreneurship

    Since 1999 I have presented 2 projects to venture capitalists in both Australia and Silicon Valley. This required making both a convincing technical argument to the technical reviewers, but also showing a business opportunity to the Venture Capitalists.

    Developed product concepts for wireless Internet security, massively parallel computers and web server admission control

    In 1994 founded an ISP. The ISP was sold in late 2007 to a larger entity.

- Technology transfer

    Led 2-man team to retrieve the Eddie admission control front-end for web servers from Sweden. This project was later commercialized by Ericsson Australia as Lodbroker.

- Network management, systems administration, application and server support

    Since 2005 I operated a multi-homed ISP focused on the businesses market. The ISP had in excess of 200 customers and provided a range of services including hosting, co-location and connectivity. The ISP was well regarded by its customers for its technical innovation, strong support and expert advice.

- Explain, present and pitch technical material

    Significant experience in working with cross disciplinary teams

    Liaised with lawyers and business people to advance commercialisation projects

EDUCATION

UNIVERSITY OF MELBOURNE, MELBOURNE BUSINESS SCHOOL ***Master of Business Administration***	Melbourne, Australia *Jan 2003 - June 2004*
MONASH UNIVERSITY ***Doctor of Philosophy, Computing and Information Technology***	Melbourne, Australia *1991 - 1996*
MONASH UNIVERSITY ***Bachelor of Science with First Class Honours in Computer Science***	Melbourne, Australia *1987 - 1990*

Prizes

- Praxa Computer Science Prize - Practical Aspects (Top Honours Student with practical focus)

WORK EXPERIENCE

Clarinet Internet Solutions August 2008 – Current

Director August 2008 – Current

Responsible for technical and business development of a small Australian ISP. After the collapse of Ustel, we repurchased most of the original business. Clarinet operates equipment in 2 datacenters and works with several providers of ADSL and wireless services.

- Achievements
  - Retrieved most of the Clarinet business, retaining the vast majority of the customers still with Ustel.
  - Re-engineered Clarinet processes to allow the business to be run from geographically distributed offices
  - Created new billing system for the business
- Technical
  - Skills: TCP/IP network administration, shell scripts, systems administration, programming in C, PHP, Perl, SQL and C++
  - Trialled virtualisation products from VMware, Virtual Iron, Citrix and Sun
  - Installed Asterisk based telephony system extended with SMS notifications
  - Setup, maintained and administered the mix of FreeBSD and Linux (Redhat and Debian) servers Clarinet used to offer hosting and control panel facilities to clients
  - Developed ZFS based server for maintaining backups

Technical Consulting to Ustel November 2007 – April 2008

Development and design tasks for the new owners of the Clarinet Internet Solutions business.

- Achievements
  - Designed new generation mail server, solved reliability problem with legacy server, added features to new billing system and created new customer front-end for the billing system that performs credit check on applicant.
- Technical
  - Skills: Knowledge of SMTP and programming in C and Javascript.
  - Developed a heartbeat daemon for the iBoot IP controlled switch. Failure of the daemon results in power cycling the legacy server solving an intermittent lock up problem.
  - Added features and customisations to a third party billing system. Seconded to Costmedia to add changes to their ASP and SQL Server based billing system as specified by Ustel.
  - Created a new front-end to the billing system using PHP, Java and XML running on a Unix host that performs a credit check on a new customer via Veda Advantage's VedaXML service before passing the customer to the 3rd party billing system.

Consultant to Access to Bandwidth Project, Australian Flexible Learning Framework, Office of Post-Compulsory Education and Training, Department of Education March 2006 – September 2007

Responsible for development of policies and technical solutions for improving broadband connectivity for the Vocational and Technical Education (VTE) sector. Project discontinued due to change of federal government

- Achievements
  - Developed a strategy and policy suggestions for linking the states TAFE networks while providing expert guidance on Internet Protocols, routing and the structure of the Australian Internet industry.
- Technical
  - Skills: Knowledge of TCP/IP including filtering, application proxies and NAT; and routing including BGP.
  - Analysed existing state network infrastructures and developed high level network design to link selected resources between states via AARNet infrastructure

CLARINET INTERNET SOLUTIONS / Blitzwave Pty Ltd January 2005 – October 2007

Director January 2005 - October 2007

Responsible for technical and business development of a small Australian ISP. Founded the ISPs sister company which holds a telecommunications carrier license. Company sold to Ustel / Ustelecom.

- Achievements
  - Refocused and redefined the Clarinet business model; founded Blitzwave to act as a wholesale carrier servicing Clarinet and other ISPs
  - Developed internal customised Linux kernel and applications for WiFi access points
  - Re-engineered business and systems processes to improve business efficiency and service delivery to customers.
- Technical
  - Skills: TCP/IP network administration, shell scripts, systems administration, programming in C, PHP, Perl, SQL and C++
  - Co-designed and operated the Clarinet network which was a multi-homed ISP taking connections from 2 providers, the VIX peering network and maintaining client machines in 2 data centres
  - Setup, maintained and administered the mix of FreeBSD and Linux (Redhat and Debian) servers Clarinet used to offer hosting and control panel facilities to clients
  - Customised the OpenWRT Linux distribution to create over 1 km links to support Blitzwave's customers. Blitzwave also used Motorola Canopy equipment to service customers.
  - Developed scripts for the Nagios monitoring system. These scripts were written in a mix of Perl, Bourne Shell and C.
  - Wrote and maintained Blitzwave's Interception Capability Plan.
  - Developed image file analysis software for detecting Spam images in C
  - Developed billing system components including a call rating system in Perl, MySQL and PHP. The system to voice call records from Clarinet's two voice call providers and computed to the cost and client charges
  - Network monitoring using TCPdump, flowd and Wireshark/Ethereal. Diagnosed customer network problems including determining link usage breakdowns, inability to reach destinations and protocol issues.
  - Redeveloped the Clarinet web site and created the Blitzwave web site using a combination of the RapidWeaver authoring tool, customised CSS, PHP and HTML. The new site provided facilities for customers to securely view their voice call bills and determine what services where available in their exchange area

RMIT UNIVERSITY May 1995 - Dec 2004

Researcher Business Faculty, Research & Development Unit (Part Time) July 2004 – Dec 2004

Researching methodologies and acting as an expert computer scientist in a multidisciplinary team of lawyers and qualitative researchers working with the Smart Internet CRC on Users and Digital Rights Management. Departed to rejoin Clarinet Internet Solutions full time.

- Achievements
  - Brought together social scientists, lawyers, scientists and engineers to focus on product design focused on the user's needs
- Technical
  - Skills: Knowledge of software development process and user interface design

Senior Research Fellow, Department of Mathematics and Statistics (Part Time) Mar 2003 – Dec 2003

Responsible for developing applications and supervising research students within the ID Group (7 members). This cross disciplinary group applied cryptography to real world problems. Departed to undertake a full time MBA.

- Achievements
  - Developed novel applications of biometrics to identify individuals and prevent identity theft
- Technical
  - Skills: TCP/IP firewalling, and knowledge of biometrics
  - Supervised a Masters student implementing a novel approach to biometrics. This approach attempted to eliminate the problem of stolen biometric templates

Senior Research Fellow, Software Engineering Research Centre (SERC) Mar 1999 - Mar 2003

Responsible for software and hardware R&D, staff and student supervision and administration. SERC was founded in July 1994 with core funding from Ericsson Australia as part of the Collaborative Information Technology Research Institute. The group (3 to 21 members) specialised in Software Engineering and Teletraffic Performance Analysis. SERC undertook applied research and development for Ericsson and other organisations, with projects sourced from both within Australia and overseas. Served in various roles including 2IC. Transferred to Maths department when the Centre closed at the end of its funding.

- Achievements
  - Developed a methodology using social science and computer science methods to solve real user problem as a technical member of a team (5 to 20 members) in the Smart Internet Technology CRC focused on the needs of Internet users.
  - Successfully developed a WiFi security add-on to prevent resource theft. Supervised developers and liaised with research office on intellectual property issues. Created a commercialisation opportunity.
  - Developed high level designs and liaised with venture capital firms and lawyers for a novel cluster based supercomputer design. Assisted in searching for venture capital to move project into hardware development for pre-production prototype.
  - Successfully rescued the Eddie/Lodbroker project by transferring the project from Sweden to Australia for Ericsson. Provided technical leadership in the transfer and set up of new team. An open source derivative Eddie (http://eddie.sourceforge.net/) survived the closure of Ericsson's 42nd Precinct.
- Technical
  - Skills: TCP/IP firewalling, software development design and implementation and programming in C, Erlang, Java, C++ and Objective-C
  - Maintained a network of over 15 FreeBSD and 3 Solaris Unix servers. Constructed the firewall and security policy. The FreeBSD boxes could be operated as a processor pool for use by the teletraffic analysts.
  - Developed a compiler for the Erlang programming language using C, Lex and YACC
  - Transferred and maintained the Eddie Load balancing web server from Ericsson Sweden. This system was written in Erlang and C.
  - Developed a firewall application that would load firewall rules when a client program authenticated itself. The Macintosh client was written in Objective-C, other clients were written in C

Lecturer, Department of Computer Science May 1995 - Mar 1999

Responsible for research and student supervision. Member of one of the largest university computer science departments in Australia; seconded to SERC half time. Member of significant committees including the Research Committee and the Software Engineering course committee.

- Achievements
  - Successfully developed admission control for web servers, which resulted in an Australian patent being filed and taken to international review stage.
  - Created challenging courses on software engineering, real time and concurrent systems which I prepared and deliverd.
  - Co-ordinated a series of successful student software projects. Some of the projects were for external clients necessitating both supervision and liaison with business people. All external student projects were successfully delivered and one resulted in the student group being employed by their sponsors.
- Technical
  - Skills: TCP/IP firewalling, software development design and implementation and programming in C, Erlang, Java, Ada, C++ and Objective-C
  - Maintained a network of over 15 FreeBSD and 3 Solaris Unix servers. Constructed the firewall and security policy. The FreeBSD boxes could be operated as a processor pool for use by the teletraffic analysts.
  - Tutored introductory programming in Java and Ada
  - Developed prototype web proxy servers and reverse proxy servers for increasing the stability of web servers and providing secured access to web resources. These were written in C and Erlang.

CLARINET INTERNET SOLUTIONS / HILINK COMMUNICATIONS June 1994 – Sept 1999

Director June 1996 – Sept 1999
Co-founder, Partner and Technical Manager June 1994 – June 1996

Founded with 2 others a small Internet service provider to support business with innovative Internet solutions; participated in the technical aspects of the setup and operations.

- Achievements
  - Setup and operated a start up ISP at the beginning of the industry in Australia. The business survived and operated until it was later sold.
  - Successfully oversaw its move from a partnership to incorporation while providing technical and management expertise
- Technical
  - Skills: TCP/IP networking and microcontroller assembly code and hardware design
  - Co-Setup one of Australia's earliest ISPs offering dialup Internet
  - Designed a serial (rs232) controlled switch box based around the 8051 microcontroller to reset modems that failed to respond to their AT reset string

WORK EXPERIENCE SUMMARY

Clarinet Internet Solutions Pty Ltd	Director	2008 -
Clarinet Internet Solutions Pty Ltd	Director	2005 - 2007
Blitzwave Pty Ltd	Director	2005 - 2007
RMIT University, Department of Mathematics & Statistics	Senior Research Fellow	2003
RMIT University, Software Engineering Research Centre	Senior Research Fellow	1999 - 2003
RMIT University, Department of Computer Science	Lecturer	1995 – 1999
HiLink Communications / HiLink Internet Pty Ltd	Partner & Technical Manager Director	1994 – 1999
Monash University, Department of Computer Science	Sessional Lecturer	1994
Monash University, Department of Computer Science	Part Time Tutor	1991 – 1994
Monash University, Department of Computer Science	Vacation Scholarship	1989

Consultancies Summary

Ustel / Ustelecom	2007 - 2008
Australian Flexible Learning Framework, Office of Post-Compulsory Education and Training, Department of Education, Access to Bandwidth Project	2006 - 2007

TECHNICAL SKILLS

Programming Languages
Expert: C, Erlang, 80x86 assembler
Knowledgeable: Java, C++, S-Plus, XML, XSLT, Javascript
Familiar with: Fortran, Cobol, 8051 assembler, Ada
Acquainted: Lisp, Scheme, Prolog

Scripting Languages
Expert: Bourne Shell, C Shell, sed, awk
Knowledgeable: Perl, TCL/Tk, PHP

Markup Languages
Expert: LaTeX, HTML
Knowledgeable: TeX, Postscript

Operating Systems
Expert: Solaris (2.5 – 2.9), FreeBSD (1.1 – 7.0), SunOS (4.1.3)
Knowledgeable: MacOS X (10.1 – 10.5.2), Linux, Solaris 10, OpenSolaris
Familiar with: WinXP, WinNT, Win95, Win3.1, W2K, VAX VMS

Network Administration
Expert: TCP/IP, Internet Security (Firewall, NAT)
Knowledgeable: UUCP, Samba, E-Mail (Sendmail), HTTP (Apache), Proxy (Squid, Fire Wall Tool Kit), Wi-Fi

PROFESSIONAL MEMBERSHIPS

- Member of the Association of Computing Machinery
- Member of the IEEE Computer Society
- Member of the Victorian FreeBSD User Group
- Member of the Systems Administrators Guild of Australia (SAGE-AU)
- Member of the Melbourne Solaris / Open Solaris User Group

ADDITIONAL INFORMATION

- Author of over 10 refereed papers and 2 text books: Assembly Language and Architecture (ISBN 0 646 28775 3) and Erlang in Real Time (ISBN 0 864 44743 4)
- Ride horses and in training for dressage.

EXHIBIT MC-2

SERC-0119
April, 2000
Issue 1

Magnus:
Software Demonstration

Maurice Castro
Robert Rendell

SERC-0119
Issue 1
April, 2000

Copies of this document may be obtained by contacting:

Director
SERC
Level 3, 110 Victoria St
Carlton, Victoria, 3053

Prepared by the Software Engineering Research Centre.

Abstract

A Radius Server is constructed in Erlang to demonstrate the feasibility of writing software for the Magnus Machine. The database component of the system is distributed and shows near linear speedup.

SERC-0119
Issue 1
April, 2000

1 Introduction

The Magnus Machine consists of a collection of Erlang Engines linked by a high speed non-blocking optical interconnect.

The Erlang Engine provides a specialised environment for the high speed execution of Erlang programs and a satellite processor is used to perform input/output operations supported in a commodity environment.

This demonstration illustrates both the ability of Erlang programs to distribute work across multiple systems and to provide a near linear 'speed up' with respect to the number of processors, and several aspects of the Erlang Engine architecture. The architectural features demonstrated include the use of a satellite processor running on a commodity operating system to interface with Internet applications, the use of a high speed optical interconnect, and the separation of the components of an application across an execution environment and an interface environment.

2 Functionality

The core of the demonstration is a Radius server implemented in Erlang. This server receives authentication requests from Radius clients and sends either Access-Accept or Access-Reject packets after verifying the username and password in a distributed database.

A Radius server implements a protocol (Radius protocol RFC 2138) for managing the authentication of users. A single database of users is queried by clients to determine if a connection is allowed and any configuration parameters required by the user. Originally Radius was developed for authenticating users on dispersed serial line and modem pools, however, more recently Radius has been applied to managing authentication for Voice over IP (VoIP) telephony.

The demonstration simulates the authentication component of a VoIP telephony system (see figure 1).

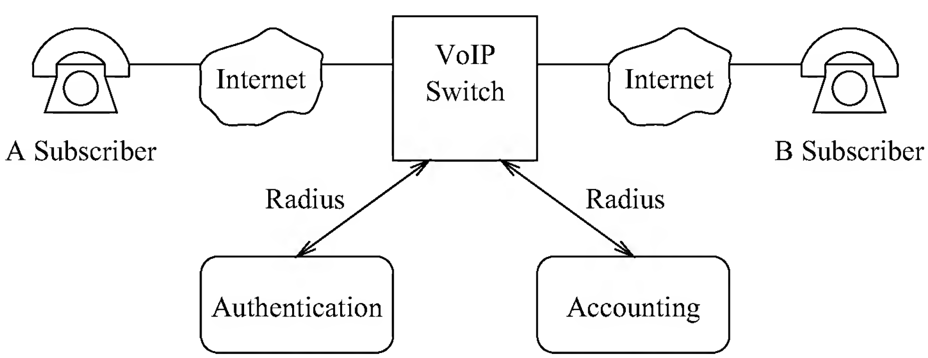

Figure 1: Voice over IP architecture

The server is instrumented to report the total number of authentication attempts, the number of successful attempts, the number of unsuccessful attempts and the time taken to lookup a name in the distributed database. Databases can be added and removed from the server while the server continues to operate.

The implementation provides about half the authentication functionality specified in RFC 2138 (Challenge-Response and allocation from a pool of available IP numbers are not implemented) nor any of the Radius accounting functions.

3 Deployment

Figure 2 shows the allocation of software to hardware used in the demonstration. The Query Generation boxes – Universe and Parallel – run Solaris and send requests to the Front End boxes selected at random from a list of possible requests. These lists include both entries which match and do not match the contents of the database used to authenticate Radius requests.

A 100 MBit/S link via an IP switch is used to deliver the requests and their associated responses.

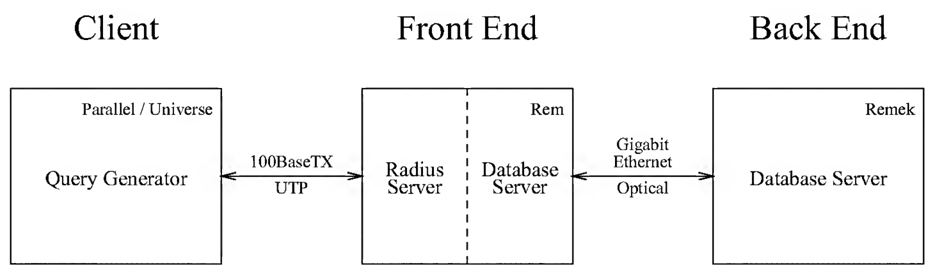

Figure 2: Hardware / Software

The Front End runs the Erlang Radius server, the display software, and 2 of the databases used in the demonstration. A 200 MHz Pentium with 32 Mb of RAM is used for the Front End.

A 1 GBit/S optical link (SysKonnect Gigabit Ethernet) is used to connect the Front End to the Back End.

The Back End runs 2 databases used in the demonstration. A 75 MHz Pentium with 16 Mb of RAM is used for the Back End.

Each database is preloaded with 10000 usernames and passwords.

4 Architecture

The demonstration is structured as 4 Erlang nodes running in Erlang emulators on top of the FreeBSD operating system (see figure 3).

The four nodes operate on 2 machines (rem and remek) and functions are assigned to each of the nodes as follows:

demo0@rem runs the Radius server which queries each of the subscribed databases, the graphical display and one of the databases

demo1@rem runs one of the databases

demo0@remek runs one of the databases

demo1@remek runs one of the databases

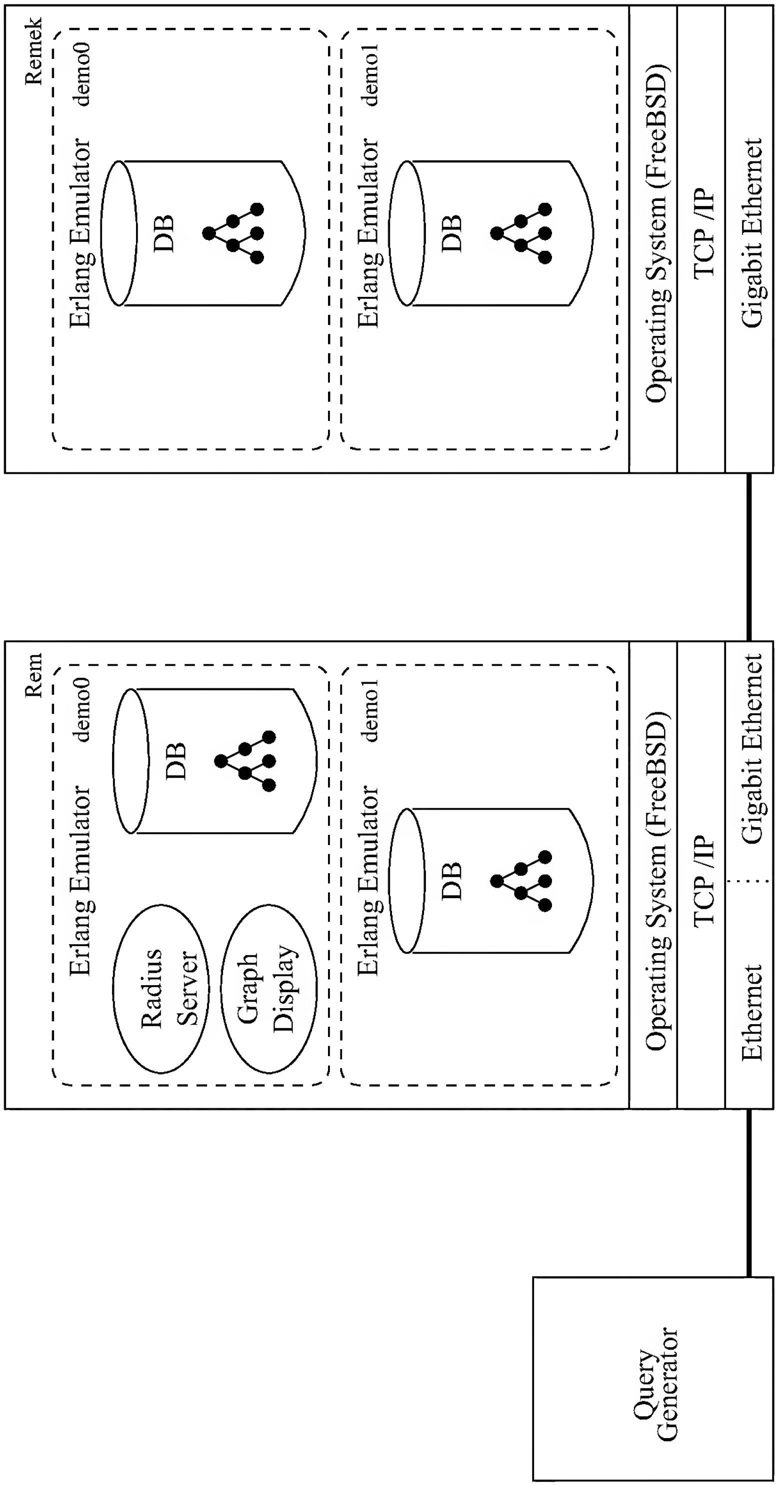

Figure 3: Detailed Architecture

SERC-0119
Issue 1
April, 2000

This architecture allows a near linear speed up to be shown. This speed up is demonstrated by varying the number of databases queried by the server. When the databases are on separate processors the 2 processors can be used to search twice the name space in close to the amount of time used by the slower processor.

5 Operation

As shown in figure 4, each time a request is received and decoded a message is sent to each of the databases to initiate a search for the username's entry. If an entry is found a positive response is sent back. If no entry is found a negative response is sent.

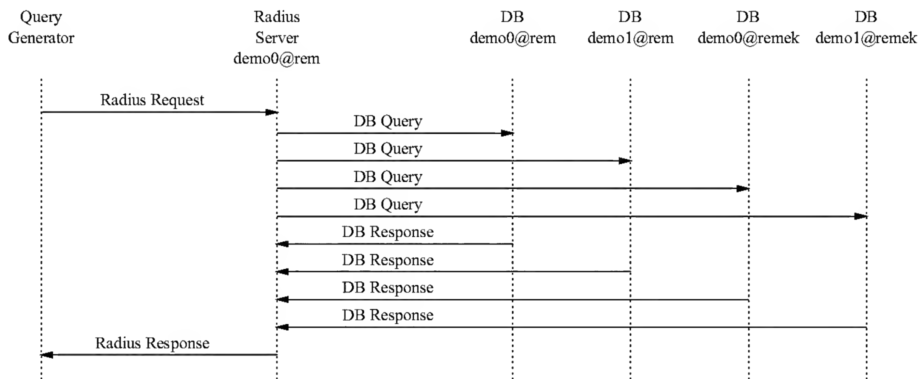

Figure 4: Message Sequence

The database is organised as an AVL tree, this structure is efficient for searches but expensive for inserts. This characteristic accurately reflects the desirable properties of an authentication service.

Although little effort was made towards writing a fault tolerant system, the failure of a database does not result in a failure of the Radius server. When the server queries the databases the server waits for either a reply from each of the queriers or a time out. This allows for the failure of the databases. When the system detects the death of a database process the process is removed from the collection of databases to be queried. A further advantage of this approach is that it allows redundancy of the databases to be simply provided. A duplicate database can be run on a separate machine containing the same data as the primary database. By including the duplicate database in the set of databases to be queried redundancy is automatically provided.

The prototype also exhibits a clean start up under load. The design easily copes with queries sent while the system is initialising. This behaviour is desirable in server applications and was achieved without changing the design. To achieve this in a conventional C or C++ implementation would require special consideration of this precondition.

6 The Display

A screen capture of the display generated by the demonstration is shown in figure 5.

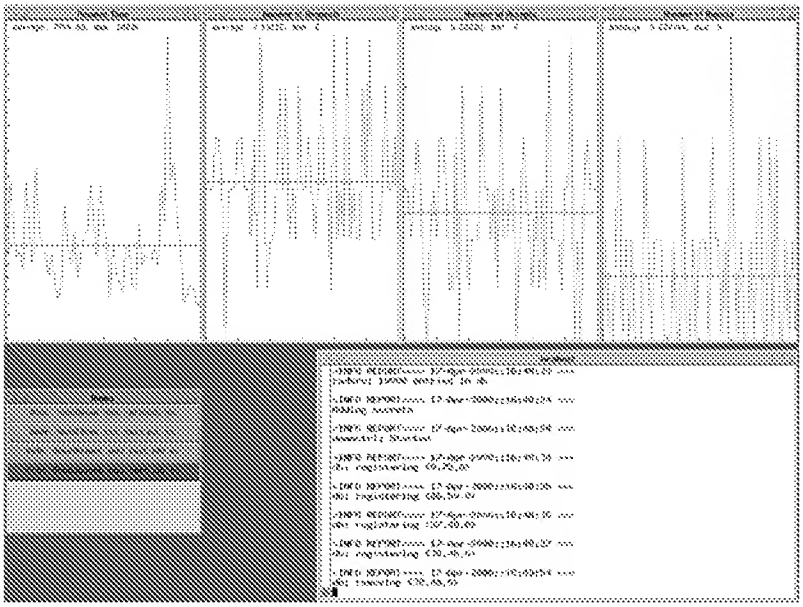

Figure 5: Demo Display

The key features of the display are the 4 graph windows shown across the top of the display. Each of the graphs contains a red line indicating the average of the displayed values, and both the maximum and minimum values displayed. From left to right these graphs show:

- the length of time required to check the database for a username (in microseconds, with tickmarks at each millisecond).
- the number of requests that have arrived in a second (with tickmarks for each five values per second)
- the number of positive responses sent in a second (with tickmarks for each five values per second)
- the number of rejected requests in a second (with tickmarks for each five values per second)

The box on the lower left hand side of the screen contains the names of the database servers. Clicking on a database subscribes the database or unsubscribes the database. A red background indicates that the database has been unsubscribed, a green background indicates that the database has been subscribed. The window on the lower right hand side contains a log of informational and error events.

SERC-0119
Issue 1
April, 2000

7 Comparison to Magnus

If this application were implemented on the Magnus architecture then figure 3 would be simplified by the removal of the operating system and the emulation environment. The memory-speed optical connection would yield a significant simplification and speed up. Figure 6 shows a potential implementation.

Further performance improvement would come from the message broadcast facility. This facility would parallelise the sending of the messages which initiate searches.

8 Program Statistics

The project was completed within 2 weeks by 2 programmers working part time. The effective effort was 1 person week.

Lines	Characters	File Name
256	9800	avl.erl
120	2920	db.erl
66	1302	dbavl.erl
136	3311	demo.erl
46	1088	democtrl.erl
187	7169	graph.erl
278	9023	md5.erl
25	580	plot.erl
404	13386	radius.erl
123	3422	radlog.erl
71	2047	radsrv.erl
110	2895	secret.erl
62	1974	server.erl
76	2153	timlog.erl
38	1149	util.erl
1998	62219	total

Of the modules listed above all were written by the 2 developers with the exception of *md5.erl* which was developed by Tony Rogvall from Ericsson. Swedish export controls prevent the use of the *crypto* module provided by OTP. The *crypto* module provides equivalent functionality to the *md5* module.

9 Program Performance

Figure 7 shows the spread of delay in looking up the names and recovering their associated parameters. The delay is measured in microseconds and the measurements were taken at a constant rate of requests (approximately 3.3 requests per second). Each pair of graphs shows the set of results gathered. The lower graph excludes a number of outlying values.

One the lower graph, each line is a degree 7 spline plotting the average of the points. The red line excludes the first 50 points to eliminate start up effects. The green dashed line includes all the points which lie within 3 standard deviations of the mean of the data set.

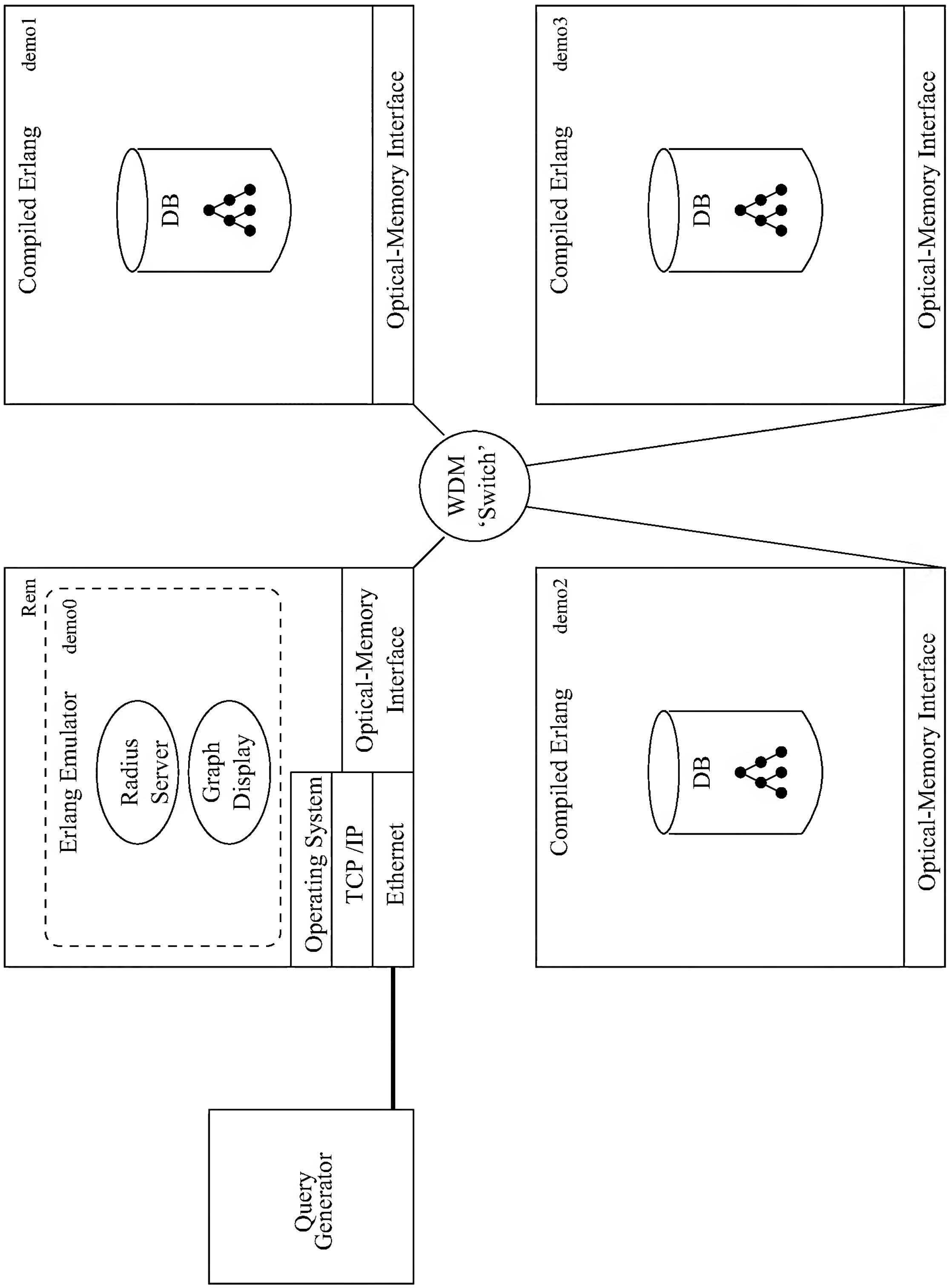

Figure 6: Magnus Implementation

The first pair of graphs are the most interesting as they show the near linear speedup offered by the Magnus architecture. The long term averages of the first 2 graphs are below 5000 and vary little. Between the first and second graphs the quantity of data searched doubles causing negligible difference in the search time. This observation characterises a near linear speedup.

The third and fourth graphs show the effects of paging and network contention. These result from doubling the load on fully loaded processors. The virtual memory mechanism employed on the test system allows the processes to continue but with severe degradation in terms of both throughput and service variability. The Magnus architecture's non-blocking switch and physical memory design would not exhibit these characteristics. This result set is merely included to show the possibility of scaling to an arbitrary number of processors with minimal code change.

The test system would support authenticating approximately 600 calls per second assuming a 3 minute hold time for each call and 40000 subscribers in the test configuration. The capacity would appear to be adequate to support a small town's switching needs on a minimal and far from optimal hardware platform.

10 Conclusion

The demonstration shows the ability of Erlang to facilitate the rapid development of distributed applications. Furthermore, these applications can be scaled by the number of available processors to provide a near linear speedup where resources are adequate to support uncontested messaging over a sufficiently fast link.

Emergent behaviour in the demonstration highlighted other advantages in using Erlang. Specifically, with no design effort the system demonstrated its ability to startup under load and to add and remove entries from the databases on-the-fly.

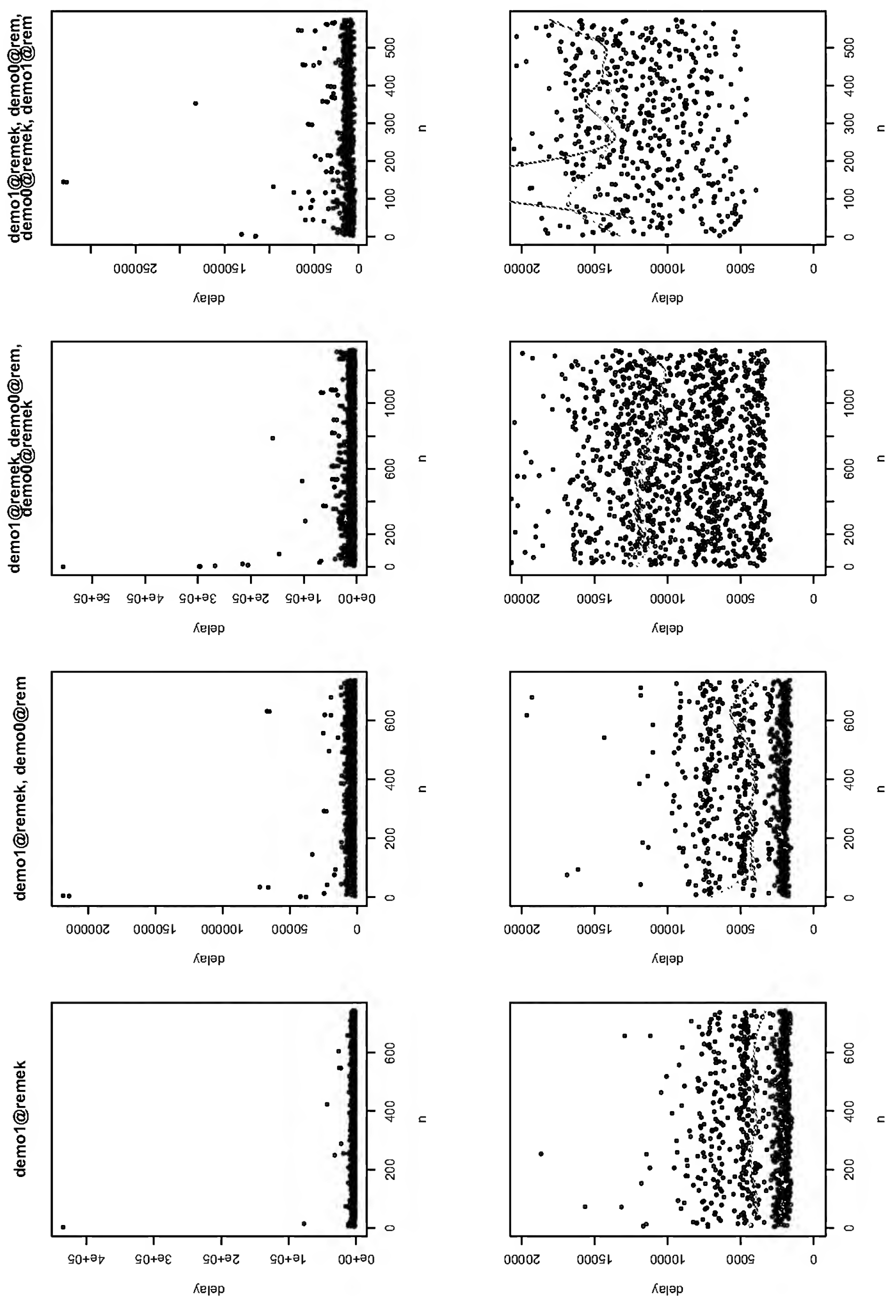

Figure 7: Spread of Delay

EXHIBIT MC-3

CLAIMS
Application Serial Number: 11/612,095

1-7. (Previously Canceled)

8. (Previously Presented) A method for constructing a scalable computer system, the method comprising:

providing a first plurality of computing nodes, and a second plurality of switching nodes, said first plurality being more numerous than said second plurality;

forming a plurality of node clusters, each node cluster comprising one of said switching nodes interconnecting a corresponding group of said computing nodes;

providing a plurality of cross-links between the node clusters;

directly connecting, using the plurality of cross-links, a corresponding plurality of pairs of node clusters selected from the plurality of node clusters in accordance with a selection process resulting in a formation of a network of said plurality of computing nodes having a higher clustering coefficient of nodes in comparison with a corresponding randomly-connected network in combination with a lower characteristic path length between the nodes in comparison with a corresponding regularly-connected network; and

wherein the steps of providing the plurality of cross-links and directly connecting the plurality of pairs of node clusters in accordance with said selection process are repeated until the resulting network comprises a small-world network having an average path length between the plurality of nodes falling within a predetermined desired range, independently of a number of said plurality of computing nodes.

9. (Previously Presented) The method of claim 8, wherein said selection process is a random or pseudo-random process.

10. (Previously Canceled)

11. (Previously Presented) The method of claim 8, wherein said predetermined range of the average path length between the plurality of nodes is less than 2.0.

12. (Previously Presented) The method of claim 11, wherein said predetermined range of the average path length between the plurality of nodes is between 1.5 and 1.7.

13. (Previously Presented) A scalable computer system constructed in accordance with the method of claim 8.

14-18. (Previously Canceled)

19. (Previously Presented) A scalable computer system comprising:

a plurality of node clusters, each said node cluster comprising a plurality of computing nodes which are interconnected via a common switch; and

a plurality of cross-links directly connecting a corresponding plurality of pairs of node clusters selected from the plurality of node clusters,

wherein the cross-links are disposed such that the pairs of node clusters form a network of said plurality of computing nodes which has a higher clustering coefficient of nodes in comparison with a corresponding randomly-connected network, in combination with a lower characteristic path length between the nodes in comparison with a corresponding regularly connected network, and

wherein said network comprises a small-world network having an average path length between the nodes falling within a predetermined desired range, independently of a number of said plurality of computing nodes.

20. (Previously Presented) The system of claim 19 wherein each said switching node comprises a non-blocking switch.

21. (Previously Presented) The system of claim 20 wherein each said non-blocking switch comprises a multi-wavelength optical switch.

22. (Previously Presented) The system of claim 19 wherein the pairs of node clusters are randomly, or pseudo-randomly selected.

23. (Previously Presented) The system of claim 19 wherein said predetermined range of the average path length between the plurality of nodes is less than 2.0.

24. (Previously Presented) The system of claim 19 wherein said predetermined range of the average path length between the plurality of nodes is between 1.5 and 1.7.

25. (Previously Presented) The system of claims 19 wherein each computing node of said plurality of computing nodes comprises a plurality of interconnected processors.